NAME
   tmux — terminal multiplexer

SYNOPSIS
   tmux [-28dqUuVv] [-f file] [-L socket-name] [-S socket-path]
       [command [flags]]

DESCRIPTION
   tmux is a terminal multiplexer; it enables a number of terminals to be accessed and controlled from a single terminal.

   tmux runs as a server-client system. A server is created automatically when necessary and holds a number of sessions, each of which may have a number of windows linked to it. A window may be split on screen into one or more panes, each of which is a separate terminal. Any number of clients may connect to a session, or the server may be controlled by issuing commands with tmux. Communication takes place through a socket, by default placed in /tmp.

   The options are as follows:
   -2 Force tmux to assume the terminal supports 256 colours.
   -8 Like -2, indicates the terminal supports 88 colours.
   -d Force tmux to assume the terminal supports default colours.
   -f file Specify an alternative configuration file. By default, tmux will look for a config file at ~/.tmux.conf. The configuration file is a set of tmux commands which are executed in sequence when the server is first started.
   -q Prevent the server sending various information messages, for example when window flags are altered.
   -L socket-name tmux stores the server socket in a directory under /tmp; the default socket is named default. This option allows a different socket name to be specified, allowing several independent tmux servers to be run. Unlike -S a full path is not necessary: the sockets are all created in the same directory.
   -S socket-path Specify a full alternative path to the server socket. If -S is specified, the default socket directory is not used and any -L flag is ignored.
   -U Unlock the server.
   -u Instruct tmux that the terminal support UTF-8.
   -V Print program version.
   -v Request verbose logging. This option may be specified multiple times for increasing verbosity. Log messages will be saved into tmux-client-PID.log and tmux-server-PID.log files in the current directory, where PID is the pid of the server or client process.

   command [flags]
   This specifies one of a set of commands used to control tmux, and described in the following sections. If no command and flags is specified, the new-session command is assumed.
QUICK START
To create a new tmux session running vi(1):

$ tmux new-session vi

Most commands have a shorter form, known as an alias. For new-session, this is new:

$ tmux new vi

Alternatively, the shortest unambiguous form of a command is accepted. If there are several options, they are listed:

$ tmux n
ambiguous command: n, could be: new-session, new-window, next-window
$

Within an active session, a new window may be created by typing C-b (ctrl-b, known as the prefix key) followed by the ‘c’ key.

Windows may be navigated with: C-b 0 (to select window 0), C-b 1 (to select window 1), and so on; C-b n to select the next window; and C-b p to select the previous window.

A session may be detached using C-b d and reattached with:

$ tmux attach-session

Typing C-b ? lists the current key bindings in the current window; up and down may be used to navigate the list or ‘Q’ to exit from it.

KEY BINDINGS

tmux may be controlled from an attached client by using a key combination of a prefix key, C-b (ctrl-b) by default, followed by a command key.

Some of the default key bindings include:

‘d’ Detach current client.
‘c’ Create new window.
‘n’ Change to next window in the current session.
‘p’ Change to previous window in the current session.
‘l’ Move to last (previously selected) window in the current session.
‘t’ Display a large clock.
‘?’ List current key bindings.

A complete list may be obtained with the list-keys command (bound to ‘?’ by default). Key bindings may be changed with the bind-key and unbind-key commands.

HISTORY

tmux maintains a configurable history buffer for each window. By default, up to 2000 lines are kept, this can be altered with the history-limit option (see the set-option command below).

MODES

A tmux window may be in one of several modes. The default permits direct access to the terminal attached to the window. The others are:

output mode
This is entered when a command which produces output, such as list-keys, is executed from a key binding.
scroll mode

This is entered with the `scroll-mode` command (bound to ‘=' by default) and permits the window history buffer to be inspected.

copy mode

This permits a section of a window or its history to be copied to a paste buffer for later insertion into another window. This mode is entered with the `copy-mode` command, bound to ‘[’ by default.

The keys available depend on whether `emacs(1)` or `vi(1)` mode is selected (see the `mode-keys` option). The following keys are supported as appropriate for the mode:

<table>
<thead>
<tr>
<th>Function</th>
<th>vi</th>
<th>emacs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start of line</td>
<td>0 or ‘’</td>
<td>C-a</td>
</tr>
<tr>
<td>Clear selection</td>
<td>Escape</td>
<td>C-g</td>
</tr>
<tr>
<td>Copy selection</td>
<td>Enter</td>
<td>M-w</td>
</tr>
<tr>
<td>Cursor down</td>
<td>j</td>
<td>Down</td>
</tr>
<tr>
<td>End of line</td>
<td>$</td>
<td>C-e</td>
</tr>
<tr>
<td>Cursor left</td>
<td>h</td>
<td>Left</td>
</tr>
<tr>
<td>Next page</td>
<td>C-f</td>
<td>Page down</td>
</tr>
<tr>
<td>Next word</td>
<td>w</td>
<td>M-f</td>
</tr>
<tr>
<td>Previous page</td>
<td>C-u</td>
<td>Page up</td>
</tr>
<tr>
<td>Previous word</td>
<td>b</td>
<td>M-b</td>
</tr>
<tr>
<td>Quit mode</td>
<td>q</td>
<td>Escape</td>
</tr>
<tr>
<td>Cursor right</td>
<td>l</td>
<td>Right</td>
</tr>
<tr>
<td>Start selection</td>
<td>Space</td>
<td>C-Space</td>
</tr>
<tr>
<td>Cursor up</td>
<td>k</td>
<td>Up</td>
</tr>
</tbody>
</table>

BUFFERS

`tmux` maintains a stack of paste buffers for each session. Up to the value of the `buffer-limit` option are kept; when a new buffer is added, the buffer at the bottom of the stack is removed. Buffers may be added using `copy-mode` or the `set-buffer` command, and pasted into a window using the `paste-buffer` command.

PANES AND LAYOUTS

Each window displayed by `tmux` may be split into one or more panes; each pane takes up a certain area of the display and is a separate terminal. A window may be split into panes using the `split-window` command.

Panels are numbered beginning from zero; in horizontal layouts zero is the leftmost pane and in vertical the topmost.

Panels may be arranged using several layouts. The layout may be cycled with the `next-layout` command (bound to `C-space` by default), the current pane may be changed with the `up-pane` and `down-pane` commands and the `rotate-window` and `swap-pane` commands may be used to swap panes without changing the window layout.

The following layouts are supported:

manual

Manual layout splits windows vertically (running across); only with this layout may panes be resized using the `resize-pane-up` and `resize-pane-down` commands.

active-only

Only the active pane is shown - all other panes are hidden.
even-horizontal
  Panes are spread out evenly from left to right across the window.

even-vertical
  Panes are spread evenly from top to bottom.

left-vertical
  A large (81 column) pane is shown on the left of the window and the remaining panes are spread from top to bottom in the leftover space to the right.

COMMANDS
This section contains a list of the commands supported by tmux. Most commands accept the optional -t argument with one of target-client, target-session or target-window. These specify the client, session or window which a command should affect. target-client is the name of the pty(4) file to which the client is connected, for example /dev/tty1. Clients may be listed with the list-clients command.

target-session is either the name of a session (as listed by the list-sessions command); or the name of a client as for target-client, in this case, the session attached to the client is used. An fnmatch(3) pattern may be used to match the session name. If a session is omitted when required, tmux attempts to use the current session; if no current session is available, the most recently created is chosen. If no client is specified, the current client is chosen, if possible, or an error is reported.

target-window specifies a window in the form session:index, for example mysession:1. The session is in the same form as for target-session. session, index or both may be omitted. If session is omitted, the same rules as for target-session are followed; if index is not present, the current window for the given session is used. When the argument does not contain a colon (:), tmux first attempts to parse it as window index; if that fails, an attempt is made to match a session or client name.

Multiple commands may be specified together as part of a command sequence. Each command should be separated by spaces and a semicolon (;); commands are executed sequentially from left to right. A literal semicolon may be included by escaping it with a backslash (for example, when specifying a command sequence to bind-key).

Examples include:

  refresh-client -t/dev/tty2

  rename-session -tfirst newname

  set-window-option -t:0 monitor-activity on

  new-window ; split-window -d

  bind-key D detach-client \; lock-server

The following commands are available:

attach-session [ -d ] [ -t target-session ]
  (alias: attach)
  Create a new client in the current terminal and attach it to a session. If -d is specified, any other clients attached to the session are detached.

  If no server is started, attach-session will attempt to start it; this will fail unless sessions are created in the configuration file.
bind-key [-r] key command [arguments]
        (alias: bind)
Bind key key to command. Keys may be specified prefixed with ‘C-’ or ‘^’ for ctrl keys, or ‘M-’
for alt (meta) keys. The -r flag indicates this key may repeat, see the repeat-time option.

break-pane [-d] [-p pane-index] [-t target-window]
Break the current pane off from its containing window to make it the only pane in a new window. If
-d is given, the new window does not become the current window.

choose-session [-t target-window]
Put a window into session choice mode, where the session for the current client may be selected
interactively from a list. This command works only from inside tmux.

choose-window [-t target-window]
Put a window into window choice mode, where the window for the session attached to the current
client may be selected interactively from a list. This command works only from inside tmux.

clock-mode [-t target-window]
Display a large clock.

command-prompt [-t target-client][template]
Open the command prompt in a client. This may be used from inside tmux to execute commands
interactively. If template is specified, it is used as the command; any %% in the template will be
replaced by what is entered at the prompt.

copy-buffer [-a src-index] [-b dst-index] [-s src-session] [-t dst-session]
        (alias: copyb)
Copy a session paste buffer to another session. If no sessions are specified, the current one is used
instead.

copy-mode [-u] [-t target-window]
Enter copy mode. The -u option scrolls one page up.

delete-buffer [-b buffer-index] [-t target-session]
        (alias: deleteb)
Delete the buffer at buffer-index, or the top buffer if not specified.

detach-client [-t target-client]
        (alias: detach)
Detach the current client if bound to a key, or the specified client with -t.

down-pane [-p pane-index] [-t target-window]
        (alias: downp)
Move down a pane.

find-window [-t target-window] match-string
        (alias: findw)
Search for match-string in window names, titles, and visible content (but not history). If only
one window is matched, it’ll be automatically selected, otherwise a choice list is shown. This com-
mand only works from inside tmux.

has-session [-t target-session]
        (alias: has)
Report an error and exit with 1 if the specified session does not exist. If it does exist, exit with 0.

kill-pane [-p pane-index] [-t target-window]
        (alias: killp)
Destroy the given pane.
kill-server
    Kill the tmux server and clients and destroy all sessions.

kill-session [-t target-session]
    Destroy the given session, closing any windows linked to it and no other sessions, and detaching all clients attached to it.

kill-window [-t target-window]
    (alias: killw)
    Kill the current window or the window at target-window, removing it from any sessions to which it is linked.

last-window [-t target-session]
    (alias: last)
    Select the last (previously selected) window. If no target-session is specified, select the last window of the current session.

link-window [-dk] [-s src-window] [-t dst-window]
    (alias: linkw)
    Link the window at src-window to the specified dst-window. If dst-window is specified and no such window exists, the src-window is linked there. If -k is given and dst-window exists, it is killed, otherwise an error is generated. If -d is given, the newly linked window is not selected.

list-buffers [-t target-session]
    (alias: lsb)
    List the buffers in the given session.

list-clients
    (alias: lsc)
    List all clients attached to the server.

list-commands
    (alias: lscm)
    List the syntax of all commands supported by tmux.

list-keys
    (alias: lsk)
    List all key bindings.

list-sessions
    (alias: ls)
    List all sessions managed by the server.

list-windows [-t target-session]
    (alias: lsw)
    List windows in the current session or in target-session.

load-buffer [-b buffer-index] [-t target-session] path
    (alias: loadb)
    Load the contents of the specified paste buffer from path.

lock-server
    (alias: lock)
    Lock the server until a password is entered.
move-window [-d] [-s src-window] [-t dst-window]
(alias: movew)
This is similar to link-window, except the window at src-window is moved to dst-window.

new-session [-d] [-n window-name] [-s session-name] [command]
(alias: new)
Create a new session with name session-name. The new session is attached to the current terminal unless -d is given. window-name and command are the name of and command to execute in the initial window.

new-window [-d] [-n window-name] [-t target-window] [command]
(alias: neww)
Create a new window. If -d is given, the session does not make the new window the current window. target-window represents the window to be created. command is the command to execute. If command is not specified, the default command is used.

The TERM environment variable must be set to “screen” for all programs running inside tmux. New windows will automatically have “TERM=screen” added to their environment, but care must be taken not to reset this in shell start-up files.

next-layout [-t target-window]
(alias: nextl)
Move a window to the next layout and rearrange the panes to fit.

next-window [-t target-session]
(alias: next)
Move to the next window in the session.

paste-buffer [-d] [-b buffer-index] [-t target-window]
(alias: pasteb)
Insert the contents of a paste buffer into the current window.

previous-window [-t target-session]
(alias: prev)
Move to the previous window in the session.

refresh-client [-t target-client]
(alias: refresh)
Refresh the current client if bound to a key, or a single client if one is given with -t.

rename-session [-t target-session] new-name
(alias: rename)
Rename the session to new-name.

rename-window [-t target-window] new-name
(alias: renamew)
Rename the current window, or the window at target-window if specified, to new-name.

resize-pane-down [-p pane-index] [-t target-window] [adjustment]
(alias: resizep-down)

resize-pane-up [-p pane-index] [-t target-window] [adjustment]
(alias: resizep-up)
Resize a pane. The adjustment is given in lines (the default is 1).

respawn-window [-k] [-t target-window] [command]
(alias: respawnw)
Reactivate a window in which the command has exited (see the remain-on-exit window option).
If \textit{command} is not given, the command used when the window was created is executed. The window must be already inactive, unless \texttt{-k} is given, in which case any existing command is killed.

\texttt{rotate-window} [\texttt{-DU}] [\texttt{-t} \textit{target-window}]
\hspace{1cm} (alias: \texttt{rotatew})

Rotate the positions of the panes within a window, either upward (numerically lower) with \texttt{-U} or downward (numerically higher).

\texttt{save-buffer} [\texttt{-a}] [\texttt{-b} \textit{buffer-index}] [\texttt{-t} \textit{target-session}] \textit{path}
\hspace{1cm} (alias: \texttt{saveb})

Save the contents of the specified paste buffer to \textit{path}. The \texttt{-a} option appends to rather than overwriting the file.

\texttt{scroll-mode} [\texttt{-u}] [\texttt{-t} \textit{target-window}]

Enter scroll mode. The \texttt{-u} has the same meaning as in the \texttt{copy-mode} command.

\texttt{select-pane} [\texttt{-p} \textit{pane-index}] [\texttt{-t} \textit{target-window}]
\hspace{1cm} (alias: \texttt{selectp})

Make pane \textit{pane-index} the active pane in window \textit{target-window}.

\texttt{select-prompt} [\texttt{-t} \textit{target-client}]

Open a prompt inside \textit{target-client} allowing a window index to be entered interactively.

\texttt{select-window} [\texttt{-t} \textit{target-window}]
\hspace{1cm} (alias: \texttt{selectw})

Select the window at \textit{target-window}.

\texttt{send-keys} [\texttt{-t} \textit{target-window}] \textit{key} \ldots
\hspace{1cm} (alias: \texttt{send})

Send a key or keys to a window. Each argument \textit{key} is the name of the key (such as \texttt{C-a} or \texttt{npage}) to send; if the string is not recognised as a key, it is sent as a series of characters. All arguments are sent sequentially from first to last.

\texttt{send-prefix} [\texttt{-t} \textit{target-window}]

Send the prefix key to a window as if it was pressed.

\texttt{server-info}
\hspace{1cm} (alias: \texttt{info})

Show server information and terminal details.

\texttt{set-buffer} [\texttt{-b} \textit{buffer-index}] [\texttt{-t} \textit{target-session}] \textit{data}
\hspace{1cm} (alias: \texttt{setb})

Set the contents of the specified buffer to \textit{data}.

\texttt{set-option} [\texttt{-gu}] [\texttt{-t} \textit{target-session}] \textit{option} \textit{value}
\hspace{1cm} (alias: \texttt{set})

Set an option. If \texttt{-g} is specified, the option is set as a global option. Global options apply to all sessions which don’t have the option explicitly set. If \texttt{-g} is not used, the option applies only to \textit{target-session}. The \texttt{-u} flag unsets an option, so a session inherits the option from the global options - it is not possible to unset a global option.

Possible options are:

\texttt{bell-action} [\texttt{any} | \texttt{none} | \texttt{current}]

Set action on window bell. \texttt{any} means a bell in any window linked to a session causes a bell in the current window of that session, \texttt{none} means all bells are ignored and \texttt{current} means only bell in windows other than the current window are ignored.
buffer-limit  number
  Set the number of buffers kept for each session; as new buffers are added to the top of the
  stack, old ones are removed from the bottom if necessary to maintain this maximum length.

default-command  command
  Set the command used for new windows (if not specified when the window is created) to
  command. The default is “exec $SHELL”.

default-path  path
  Set the default working directory for processes created from keys, or interactively from the
  prompt. The default is the current working directory when the server is started.

history-limit  lines
  Set the maximum number of lines held in window history. This setting applies only to new
  windows - existing window histories are not resized and retain the limit at the point they
  were created.

lock-after-time  number
  Lock the server after number seconds of inactivity. The default is off (set to 0). This has
  no effect as a session option; it must be set as a global option using -g.

message-attr attributes
  Set status line message attributes, where attributes is either default or a comma-
delimited list of one or more of: bright (or bold), dim, underscore, blink, reverse, hidden, or italics.

message-bg colour
  Set status line message background colour, where colour is one of: black, red,
green, yellow, blue, magenta, cyan, white or default.

message-fg colour
  Set status line message foreground colour.

prefix key
  Set the current prefix key.

repeat-time number
  Allow multiple commands to be entered without pressing the prefix-key again in the speci-
dified number milliseconds (the default is 500). Whether a key repeats may be set when it is
  bound using the -r flag to bind-key. Repeat is enabled for the default keys of the
  up-pane, down-pane, resize-pane-up, and resize-pane-down commands.

set-remain-on-exit [on | off]
  Set the remain-on-exit window option for any windows first created in this session.

set-titles [on | off]
  Attempt to set the window title using the \e[2;...007 xterm code and the terminal appears to
  be an xterm. This option is enabled by default. Note that elinks(1) will only attempt to
  set the window title if the STY environment variable is set.

status [on | off]
  Show or hide the status line.

status-attr attributes
  Set status line attributes.

status-bg colour
  Set status line background colour.
**status-fg** *colour*
Set status line foreground colour.

**status-interval** *interval*
Update the status bar every *interval* seconds. By default, updates will occur every 15 seconds. A setting of zero disables redrawing at interval.

**status-keys** [**vi** | **emacs**]
Use **vi**(1) - or **emacs**(1) -style key bindings in the status line, for example at the command prompt. Defaults to emacs.

**status-left** *string*
Display *string* to the left of the status bar. *string* will be passed through **strftime**(3) before being used. By default, the session name is shown. *string* may contain any of the following special character pairs:

- **#(command)**: First line of command’s output
- **#H**: Hostname of local host
- **#S**: Session name
- **#T**: Current window title
- **##**: A literal ‘#’

Where appropriate, these may be prefixed with a number to specify the maximum length, for example **#24T**.

**status-left-length** *length*
Set the maximum *length* of the left component of the status bar. The default is 10.

**status-right** *string*
Display *string* to the right of the status bar. By default, the date and time will be shown. As with **status-left**,*string* will be passed to **strftime**(3) and character pairs are replaced.

**status-right-length** *length*
Set the maximum *length* of the right component of the status bar. The default is 40.

**set-password** [**-c**] *password*
(alias: **pass**)
Set the server password. If the **-c** option is given, a pre-encrypted password may be specified. By default, the password is blank, thus any entered password will be accepted when unlocking the server (see the **lock-server** command). To prevent variable expansion when an encrypted password is read from a configuration file, enclose it in single quotes ('').

**set-window-option** [**-gu**] [**-t** *target-window*] *option* *value*
(alias: **setw**)
Set a window-specific option. The **-g** and **-u** flags work similarly to the **set-option** command. Supported options are:

**aggressive-resize** [**on** | **off**]
Aggressively resize the chosen window. This means that **tmux** will resize the window to the size of the smallest session for which it is the current window, rather than the smallest session to which it is attached. The window may resize when the current window is changed on another sessions; this option is good for full-screen programs which support SIGWINCH and poor for interactive programs such as shells.
automatic-rename [on | off]
Control automatic window renaming. When this setting is enabled, tmux will attempt - on supported platforms - to rename the window to reflect the command currently running in it. This flag is automatically disabled for an individual window when a name is specified at creation with new-window or new-session, or later with rename-window. It may be switched off globally with:

```
set-window-option -g automatic-rename off
```

clock-mode-colour colour
Set clock colour.

clock-mode-style [12 | 24]
Set clock hour format.

force-height height

force-width width
Prevent tmux from resizing a window to greater than width or height. A value of zero restores the default unlimited setting.

mode-attr attributes
Set window modes attributes.

mode-bg colour
Set window modes background colour.

mode-fg colour
Set window modes foreground colour.

mode-keys [vi | emacs]
Use vi(1) - or emacs(1) -style key bindings in scroll and copy modes. Key bindings default to emacs.

monitor-activity [on | off]
Monitor for activity in the window. Windows with activity are highlighted in the status line.

remain-on-exit [on | off]
A window with this flag set is not destroyed when the program running in it exits. The window may be reactivated with the respawn-window command.

utf8 [on | off]
Instructs tmux to expect UTF-8 sequences to appear in this window.

window-status-attr attributes
Set status line attributes for a single window.

window-status-bg colour
Set status line background colour for a single window.

window-status-fg colour
Set status line foreground colour for a single window.

xterm-keys [on | off]
If this option is set, tmux will generate xterm(1) -style function key sequences; these have a number included to indicate modifiers such as shift, meta or ctrl.
show-buffer [-b buffer-index] [-t target-session]  
(alias: showb)  
Display the contents of the specified buffer.

show-options [-t target-session] option value  
(alias: show)  
Show the currently set options. If a target-session is specified, the options for that session are shown; otherwise, the global options are listed.

show-window-options [-t target-window] option value  
(alias: show)  
List the current options for the given window.

source-file path  
(alias: source)  
Execute commands from path.

split-window [-d] [-l lines | -p percentage] [-t target-window] [command]  
(alias: splitw)  
Creates a new window by splitting it vertically. The -l and -p options specify the size of the new window in lines, or as a percentage, respectively. All other options have the same meaning as in the new-window command.

A few notes with regard to panes:
1. If attempting to split a window with less than eight lines, an error will be shown.
2. If the window is resized, as many panes are shown as can fit without reducing them below four lines.
3. The minimum pane size is four lines (including the separator line).
4. The panes are indexed from top (0) to bottom, with no numbers skipped.

start-server  
(alias: start)  
Start the tmux server, if not already running, without creating any sessions.

suspend-client [-c -target-client]  
(alias: suspenc)  
Suspend a client by sending SIGTSTP (tty stop).

swap-pane [-dDU] [-p src-index] [-t target-window] [-q dst-index]  
(alias: swapp)  
Swap two panes within a window. If -U is used, the pane is swapped with the pane above (before it numerically); -D swaps with the pane below (the next numerically); or dst-index may be give to swap with a specific pane.

swap-window [-d] [-s src-window] [-t dst-window]  
(alias: swapw)  
This is similar to link-window, except the source and destination windows are swapped. It is an error if no window exists at src-window.

switch-client [-c target-client -t target-session]  
(alias: switchc)  
Switch the current session for client target-client to target-session.

unbind-key key  
(alias: unbind)  
Unbind the key bound to key.
unlink-window [-t target-window]
   (alias: unlinkw)
   Unlink target-window. A window may be unlinked only if it is linked to multiple sessions - windows may not be linked to no sessions.

up-pane [-p pane-index] [-t target-window]
   (alias: upp)
   Move up a pane.

FILES
   ~/.tmux.conf
      default tmux configuration file

SEE ALSO
   pty(4)

AUTHORS
   Nicholas Marriott (nicm@users.sourceforge.net)